

Laboratorio de Computación II

TRABAJO PRÁCTICO N° 3 - Parte 2

Presentación del trabajo

La aplicación sirve para obtener información sobre la producción de un establecimiento ganadero, donde se produce leche y se presta el servicio de engorde.

- **Sección “Existencias ganaderas”**

En la sección denominada “Existencias ganaderas” se puede consultar el listado completo de bovinos, con su identificación única, uso (tambo o engorde), fecha de ingreso, sexo y raza.

La lista de bovinos se carga desde la base de datos ESTABLECIMIENTO. Para crear base de datos, las tablas y su contenido se debe utilizar el script adjunto.

El usuario puede agregar bovinos (a menos que ya haya un bovino cargado con el mismo identificador, que debe ser único), eliminarlos o modificar sus datos (excepto el identificador único). Tampoco se pueden agregar bovinos machos destinados a la actividad de tambo. Al ir guardando los cambios se van agregando los registros a la mencionada base de datos.

Asimismo, el usuario puede exportar los datos, tanto en formato .json, como .xml y en .txt.

- **Sección “Información productiva”**

En la sección denominada “Existencias ganaderas” se puede visualizar la siguiente información y estadísticas sobre el establecimiento:

Datos sobre el stock ganadero:

- Distribución total por uso

Datos sobre la producción lechera:

- Cabezas destinadas al tambo

- Producción de leche en el último año: Se obtiene un promedio diario para cada bovino destinado a tambo de manera aleatoria (entre 20 y 50 litros diarios). Luego se calculan los

últimos 365 días de permanencia para los animales que tienen mas de 1 año en los establecimientos, o todos los días para los que tienen menos de un año. De esta manera, se obtiene la producción total en el último año calendario.

- Raza más productiva para tambo.
- Animal más productivo para tambo.

Datos sobre la producción ganadera:

- Cabezas destinadas al engorde
- Producción de leche en el último año: Se obtiene un promedio diario para cada bovino destinado a engorde de manera aleatoria (entre 1 y 2 kilos diarios). Luego se calculan los últimos 365 días de permanencia para los animales que tienen mas de 1 año en los establecimientos, o todos los días para los que tienen menos de un año. De esta manera, se obtiene la producción total en el último año calendario.
- Raza más productiva para engorde.
- Animal más productivo para engorde.

Además, se pueden consultar las estadísticas históricas del establecimiento y se puede exportar la información obtenida tanto a xml, como a json y a txt.

Aplicación de temas aprendidos (clases 10 a 20)

• Clase 10: Excepciones

La aplicación valida datos a fin de evitar que se generen excepciones y el programa se cierre abruptamente, sin poder brindar una explicación al usuario.

Para los casos en los que es más probable que se generen excepciones, se realizó el correspondiente manejo de las mismas mediante bloques try catch.

En el caso de manejo de archivos, siempre se realiza un bloque catch para excepciones de tipo FileNotFoundException y DirectoryNotFoundException que lanza una excepción con un mensaje afín y más humano a fin de atraparlo desde el Program.cs o el Form que se esté utilizando. Al exportar datos, como la ruta del archivo de destino es un dato ingresado por el usuario también se valida que dicho campo no este vacío.

En el caso de conexiones con la base de datos, todos los métodos que interactúan con la base de datos tienen manejo de excepciones generales y específicas, como `NullReferenceException` o `SQLException` numero 2627, que es lanzada cuando la clave primaria ya se encuentra utilizada. También cuentan con un bloque `finally` que se encarga de cerrar la conexión con la base, haya existido una excepción o no.

Además, se crearon excepciones propias `AnimalExistenteException`, la cual se lanza en el caso de que el bovino que se quiere agregar a la lista ya se encuentre en la misma (tenga el mismo identificador) y `AnimalInexistenteException`, la cual se lanza en el caso contrario.

- **Clase 11: Pruebas unitarias**

Se cuenta con un proyecto de Test Unitarios, en el cual se realizan múltiples pruebas a los diferentes métodos de la biblioteca de clases, la cuales esperan tanto valores de variables como de excepciones.

Ejemplos:

`CalcularPermanenciaEnDias_RecibeAnimalIngresadoHoy_DeberiaDevolver1()`

`AgregarAnimal_RecibeAnimalQueYaEstaEnLaLista_DeberiaDevolverAnimalExistenteException()`

- **Clase 12: Tipos genéricos**

Se cuenta con una clase estática `Archivos` que recibe un tipo de referencia, al cual se va a serializar a un archivo que puede ser xml, json o txt. A ese tipo tambien se deserializa desde json.

- **Clase 13: Interfaces**

Se creó la interfaz `IBovino`, que define los métodos que deben ser implementados para estos animales. Esta es utilizada por la clase `Bovino`.

- **Clase 14: Sistemas de archivos y serialización**

Se utilizan varios métodos de las clases utilizadas para manejar el sistema de archivos.

Ejemplos: Para obtener el directorio actual al momento de definir el path se utiliza `Directory.GetCurrentDirectory()`, para armar una ruta se utiliza `Path.Combine()` y para verificar si un archivo existe se utiliza `File.Exists()`.

Tanto el listado de bovinos como la información obtenida del análisis de datos se pueden exportar en los formatos json, xml y txt, utilizando los métodos de la clase genérica `Archivos.cs`.

- **Clase 15: Introducción a bases de datos y SQL**

Utilizando SQL Server Management Studio, se ha creado una base de datos ESTABLECIMIENTO para almacenar información sobre el establecimiento y lograr la persistencia de datos.

La misma consta de 2 tablas:

- La tabla BOVINOS almacena la lista de bovinos y tiene como Primary Key el número de identificación. Este atributo también es la columna de identidad, por lo que se incrementa automáticamente a partir del número 1000.
- La tabla ESTADISTICAS tiene como Primary Key la fecha de la estadística, ya que sólo puede haber un reporte por día. Si se carga un reporte adicional en el mismo día en vez de dar de alta un registro, se actualiza el existente. Además, tiene como Foreign Key el número de identificación del bovino de tambo más productivo de ese día, ya que se va a mostrar su información extrayendo la información desde la tabla BOVINOS utilizando un INNER JOIN.

- **Clase 16: Conexión a bases de datos**

Se ha conectado la aplicación con la base de datos para poder manipularla desde la aplicación. Para ello, se instaló el paquete NuGet system.data.sqlclient en la solución.

Esto se ha implementado a partir de dos clases estáticas BovinoDAO y EstadisticasDAO, que tienen todos los métodos necesarios para realizar alta, bajas y modificaciones (con métodos ExecuteNonQuery()) y traer información de la base de datos, ya sea de registros individuales o listas de registros (ExecuteReader()).

- **Clase 17: Delegados y expresiones lambda**

Delegados: Se usaron delegados tanto como para implementar eventos como hilos.

Expresiones lambda: Para calcular el bovino más productivo, se ordena la lista de StockGanadero mediante el método Sort(), al cual se le pasa una expresión lambda que compara la producción promedio diaria de dos bovinos y devuelve un entero según el resultado de esa comparación.

- **Clase 18: Hilos**

Se utiliza un hilo para simular el cálculo de estadísticas, lo cual se muestra en una barra de progreso en el `FrmEstablecimientoGanadero`. Las misma va aumentando y cuando llega al 100% se habilita el botón de la sección “Información del establecimiento” y el label muestra “Cálculo completado”. El tiempo en el que aumenta el porcentaje se genera aleatoriamente, cada 5 puntos de porcentaje aumenta de entre 50 a 250 milisegundos.

Como son subprocesos paralelos, mientras tanto se puede consultar el formulario del stock ganadero o cerrar el programa.

- **Clase 19: Eventos**

En la clase `Establecimiento` se declara un evento `CambioStock` en base al delegado `NotificadorCambioStock`, que no recibe ningún parámetro y devuelve void.

Este evento es de tipo estático y se lanza cuando el `StockGanadero` sufre alguna modificación (se agrega, modifica o eliminar un bovino).

En el constructor de `FrmStockGanadero` se asocia ese evento al método `manejadorActualizarDataGridView` por lo que no es necesario llamarlo cada vez que actualicemos el stock, ya que se va a invocar automáticamente cuando se lance el evento. También se le asocia el evento `InformarCambio` que informa al usuario que la operación ha sido exitosa.

- **Clase 20: Métodos de extensión**

Dentro de la clase estática `CalculadoraDiasPermanencia`, se ha creado un método estático llamado `CalcularPermanenciaEnDias` que extiende la clase `DateTime` a fin de calcular cuantos días pasaron desde la fecha de ingreso del bovino hasta la fecha actual.

Dado que es de extensión, si bien es un método estático que recibe un objeto de tipo `DateTime`, al momento de llamarlo también se puede utilizar como método de instancia: `fecha.CalcularPermanenciaEnDias();`